1 SUMMARY

Evaluates a spline $S(x)$ **of degree** $k-1$ **and its derivatives** using the B-spline representation of $S(x)$. Specifically, given knots and coefficients $a_1,...,a_{m+k}$ in the representation

$$S(x) = \sum_{i=1}^{m+k} a_i N_{k,i}(x), \qquad m \geq 0, \qquad k \geq 1,$$

this subroutine computes the values of

$$\frac{d^{j-1}}{dx^{j-1}} S(x), \qquad j=1,2,....,r, \qquad r \leq k$$

at a specified point x.

The method is based on De Boor, 'On Calculating with B-splines', J. App. Theory, **6**, (1972).

ATTRIBUTES — **Version:** 1.0.0. **Types:** `TG03A`, `TG03AD`. **Original date:** January 1977. **Origin:** P.W.Gaffney, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument lists

The single precision version:

```
CALL TG03A(K,MPK,A,T,MP2K,WK,IW,XVALUE,ID,S)
```

The double precision version:

```
CALL TG03AD(K,MPK,A,T,MP2K,WK,IW,XVALUE,ID,S)
```

`K` is an `INTEGER` variable which must be set by the user to the order, k, of the spline $S(x)$. The value of k must be greater than or equal to 1. This argument is not altered by the subroutine.

`MPK` is an `INTEGER` variable which must be set by the user to the number, $m+k$, of coefficients a_i, $i=1,2,...,m+k$. The value of $m+k$ must be greater than or equal to k. This argument is not altered by the subroutine.

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least $m+k$, which must be set by the user to the values of the coefficients $a_1,..., a_{m+k}$. This argument is not altered by the subroutine.

`T` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least $m+2k$. On entry to the subroutine `T` must contain the $m+2k$ knots t_i, $i=1,...,m+2k$, which are required in order to write $S(x)$ as a linear combination of $m+k$ B-splines (see section 3). The knots t_i must be in ascending order, $t_1 \leq t_2 \leq ... \leq t_{m+2k}$ and they must also satisfy the inequalities $t_i < t_{i+k}$ $i=1,...,m+k$. This argument is not altered by the subroutine.

`MP2K` is an `INTEGER` variable which must be set by the user to the length of the array `T`. This argument is not altered by the subroutine.

`WK` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least $2k$, which is used as workspace.

`IW` is an `INTEGER` variable which must be set by the user to the length of the array `WK`. This argument is not altered by the subroutine.

`XVALUE` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set by the user to the value of the argument x at which $S(x)$ and its derivatives are to be computed. This argument is not altered by the subroutine.

`ID` is an `INTEGER` variable which must be set by the user. The subroutine computes the values $d^{j-1}S(x)/dx^{j-1}$,

j=1,...,r where $r = \max[1, \min(\texttt{ID}, k)]$.

`S` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least `ID`. On exit from the subroutine `S(j)` contains the values of $d^{j-1}S(x)/dx^{j-1}$, j=1,...,r where $r = \max[1, \min(\texttt{ID}, k)]$.

2.2 Checks on the input parameters

The restrictions imposed on the values of `K` and `MPK` are checked by the subroutine. If these restrictions are not satisfied an error diagnostic is printed, an error return flag, `IFAIL`, is set (see section 2.3), and a return is made to the calling program.

2.3 The common area and diagnostic messages

The subroutine uses a common area which the user may also reference. To do this the calling program will require a `COMMON` statement of the form:

The single precision version:

```
COMMON/TG03B/ LP,IFAIL
```

The double precision version:

```
COMMON/TG03BD/ LP,IFAIL
```

`LP` is an `INTEGER` variable which is defaulted to the value 6. All error messages appear on the unit number whose value appears in `LP` (`LP`≥`1`). The user may suppress the printing by setting `LP=0`.

`IFAIL` is an `INTEGER` variable which is set to an error return flag. On exit from the subroutine it has one of the following values:

- 0 successful entry,
- 1 `K` < `1`,
- 2 `MPK` < `K`.

2.4 Motivation

If $S(x)$ is a spline of degree k–1 with m given knots, η_i, i=1,...,m, where $-\infty < \eta_1 \le \eta_2 \le ... \le \eta_m < \infty$, then $S(x)$ has m+k linear parameters. Therefore, it can be expressed as a linear combination of m+k linearly independent normalised B-splines of degree k–1. In order to do this it is necessary to introduce an additional 2k knots t_j, j=1,...,k,m+k+1,...,m+2k such that

$$t_1 \le t_2 \le ... \le t_k < \eta_1 \tag{2.4.1}$$

and

$$\eta_m < t_{m+k+1} \le t_{m+k+2} \le ... \le t_{m+2k}. \tag{2.4.2}$$

Intermediate values of t_j are defined by the equations

$$t_{k+j} = \eta_j, \qquad j=1,...,m. \tag{2.4.3}$$

Then, for x in the range $t_k \le x \le t_{m+k+1}$, $S(x)$ can be expressed in the form

$$S(x) = \sum_{i=1}^{m+k} a_i N_{k,i}(x) \tag{2.4.4}$$

where the function $N_{k,i}(x)$ is the normalised B-spline of degree k–1 with knots at t_i,...,t_{i+k}. The purpose of this subroutine is to compute the derivatives $d^{j-1}S(x)/dx^{j-1}$, $1 \le j \le k$, at a given point x. If $x < t_k$ or $x > t_{m+k+1}$ then these derivatives are set to zero.

In order to use the subroutine the user must supply the complete set of knots t_i, i=1,...m+2k, and also the coefficients a_1,...,a_{m+k} in expression (2.4.4).

3 GENERAL INFORMATION

Use of common: The subroutine uses a common area `TG03B/BD` (see § 2.3).

Workspace: The user provides a workspace array `WK` of size at least $2k$.

Other routines called directly: None.

Input/output: Diagnostic printing is given on stream number `LP`, and may be suppressed, see §2.3.

Restrictions: $k \geq 1$, $m+k \geq k$, $t_1 \leq t_2 \leq \ldots \leq t_{m+2k}$, $t_i < t_{i+k}$, $i=1,\ldots,m+k$.

4 METHOD

To compute the derivatives

$$S^{(j-1)}(x) \equiv \frac{d^{j-1}}{dx^{j-1}} \sum_{i=1}^{m+k} a_i N_{k,i}(x), \qquad 1 \leq j \leq k. \tag{4.1}$$

for a value of x in the range $t_k \leq x \leq t_{k+m+1}$ we use the formula of De Boor (1972)

$$S^{(j-1)}(x) = \sum_{i=\mathtt{JINT}-k+j}^{\mathtt{JINT}} a_{i,j} N_{k-j+1}(x) \tag{4.2}$$

where the constants $a_{i,j}$ are calculated from the recurrence relation

$$a_{i,j} = (k-j+1)\left(\frac{a_{i,j-1} - a_{i-1,j-1}}{t_{i+k-j+1} - t_i}\right), \qquad j \geq 2 \tag{4.3}$$

starting with

$$a_{i,1} = a_i. \tag{5.4}$$

The integer `JINT,` in formula (4.2), is defined by the inequalities

$$t_{\mathtt{JINT}} \leq x \leq t_{\mathtt{JINT+1}}. \tag{5.5}$$

Reference

De Boor, C. (1972). “On Calculating with B-splines”. J. App. Theory, 6, 50-62.

5 EXAMPLE OF USE

We present an example of a situation where `TG03A` may be used. Suppose we are given values $f_1,\ldots,f_n$ of a function $f(x)$ at n distinct points $x_1 < x_2 < \ldots < x_n$, and we wish to approximate $f(x)$ by a quadratic spline $S(x)$, which has m knots $\eta_1 \leq \eta_2 \leq \ldots \leq \eta_m$ in the open interval (x_1, x_n), and which passes through the values $f_1,\ldots,f_n$. Furthermore, suppose that the eventual aim of the approximation is to tabulate $S(x)$ and its derivative $S'(x)$. Then, in order to obtain a unique spline of degree 2 which interpolates n arbitrarily prescribed values, $f_1,\ldots,f_n$, the value of m must be n–3. Moreover, the knots η_i, $i=1,\ldots,n-3$ must satisfy the inequalities $x_i < \eta_i < x_{i+3}$, $i=1,\ldots,n-3$. Under these conditions the spline $S(x)$ may be written in the form

$$S(x) = \sum_{i=1}^{n} a_i N_{3,i}(x),$$

where the constants $a_1,\ldots,a_n$ are determined uniquely from the conditions

$$S(x_j) = \sum_{i=1}^{n} a_i N_{3,i}(x_j) = f_j, \qquad j=1,\ldots,n.$$

Once a suitable set of knots η_i, $i=1,\ldots n-3$ has been chosen the values of $a_1,\ldots,a_n$ may be computed by using subroutine TB06A. In order to tabulate $S(x)$ and $S'(x)$ at values of the argument x we may use subroutine TG03A. The complete set of knots t_i, i=1,...,n+3 (see §2.4), are provided by TB06A in positions WK(1),...,WK(n+3) of the workspace array WK.

The Fortran code which is required to obtain $S(x)$, and then tabulate $S(x)$ and $S'(x)$ might be as follows.

```
      REAL X(100)                          Data points x_i, i=1,...,n (n≤100).
      REAL SX(1000),SIX(1000)              Arrays for tabulated values.
      REAL A(100)                          Coefficients a_1,...,a_n.
      REAL ETA(97)                         Knots η_i, i=1,...,n–3.
      REAL S(2)                            Array for derivatives.
      REAL AN(100,3),WK(209)               Workspace.
      INTEGER IL(100)                      Integer work space.
      READ(5,100) N                        Get n the number of data points.
      READ(5,200) X,F                      Input data points and function values x_i, f_i, i=1,...,n.
      K=3                                  set order, k, of spline S(x).
      NM3=N-K                              Set number of knots.
      READ(5,300) ETA                      Input the knots η_i, i=1,...,n–3.
      ISW=2*N+3*K                          Dimension of workspace WK.
   Compute coefficients a_1,..., a_n.
      CALL TB06A(N,X,F,K,NM3,ETA,IL,AN,ISW,WK,A)
      H=(X(N)–X(1))/999.0                  Increment for tabulation points.
      ID=2                                 Set for calculating S(x) and dS(x)/dx.
      MP2K=N+K                             Total number of knots t_i.
      MP2KP1=MP2K+1                        Set for workspace.
      IW=2*K                               Set for workspace.
   Tabulate S(x) and S^(1)(x) at the points x_i = x_i + (i–1)h i=1,...,1000 where h = (x_n – x_1)/999.
      DO 10 I=1,1000
        XVALUE=X(1)+FLOAT(I-1)*H
        CALL TG03A(K,NM3,A,WK(1),MP2K,WK(MP2KP1),IW,XVALUE,ID,S)
        SX(I)=S(1)                         Store the results in
        SIX(I)=S(2)                        SX and SIX.
   10 CONTINUE
      STOP
      END
```